WHAT IS CLAIMED IS:

1. A method of file system protection for a resource-sparing operating system (OS) image, comprising the steps of:

loading the image into random access memory (RAM), the image including a catalog file embedded therein;

creating a first hash of the image;

extracting a second hash of the image from the catalog file; and

blocking the use of the image to boot the computing device when the first hash and the second hash do not match.

2. The method of claim 1, wherein the step of blocking the use of the image to boot the computing device when the first hash and the second hash do not match comprises the step of determining an operational mode of the computing device is set to a run mode of operation.

3. The method of claim 2, wherein the step of blocking the use of the image to boot the computing device when the first hash and the second hash do not match is bypassed when the step of determining the operational mode of the computing device is set to a test mode of operation, the method further comprising the step of loading the image into a flash memory of the computing device.

4. The method of claim 1, further comprising the steps of validating a signature certification of the catalog file, and blocking the use of the image to boot the computing device when the signature certification of the catalog file cannot be validated.

5. The method of claim 4, wherein the step of blocking the use of the image to boot the computing device when the signature certification of the catalog file cannot be validated comprises the step of determining an operational mode of the computing device is set to a run mode of operation.

6. The method of claim 5, wherein the step of blocking the use of the image to boot the computing device when the signature certification of the catalog file cannot be validated is bypassed when the step of determining the operational mode of the computing device is set to a test mode of operation, the method further comprising the step of loading the image into a flash memory of the computing device.

7. The method of claim 1, further comprising the steps of extracting first make and model attributes from the catalog file, comparing the first make and model attributes from the catalog file with second make and model attributes of the computing device, and blocking the use of the image to boot the computing device when the first make and model attributes do not match the second make and model attributes.

8. The method of claim 7, wherein the step of blocking the use of the image to boot the computing device when the first make and model attributes do not match the second make and model attributes comprises the step of determining an operational mode of the computing device is set to a run mode of operation.

9. The method of claim 8, wherein the step of blocking the use of the image to boot the computing device when the first make and model attributes do not match the second make and model attributes is bypassed when the step of determining the operational mode of the computing device is set to a test mode of operation, the method further comprising the step of loading the image into a flash memory of the computing device.

10. The method of claim 1, further comprising the step of booting the computing device from a prior image already loaded in flash memory of the computing device.

11. A method of file system protection for a resource-sparing operating system (OS) image, the image including a catalog file embedded therein, comprising the steps of:

examining the catalog file and the image to determine if the image is a properly released image; and

blocking use of the image to boot the computing device when the step of examining determines that the image is not a properly released image.

12. The method of claim 11, wherein the step of examining is initiated upon a request to update the image, the method further including the step of loading an update image into random access memory (RAM).

13. The method of claim 11, wherein the step of examining is initiated upon a reset of the device.

14. The method of claim 11, wherein the step of examining comprises the steps of:

creating a first hash of the image;

extracting a second hash of the image from the catalog file; and

comparing the first hash and the second hash, and wherein a mismatch provides an indication that the image is not a properly released image.

15. The method of claim 14, further comprising the step of determining an operational mode of the device, and wherein the step of blocking the use of the image to boot the device is bypassed when the operational mode is set to test mode.

16. The method of claim 11, wherein the step of examining comprises the steps of:

extracting a signature certification from the catalog file;

validating the signature certification; and

wherein failure of the step of validating the signature certification provides an indication that the image is not a properly released image.

17. The method of claim 16, further comprising the step of determining an operational mode of the device, and wherein the step of blocking the use of the image to boot the device is bypassed when the operational mode is set to test mode.

18. The method of claim 11, wherein the step of examining comprises the steps of:

extracting first make and model attributes from the catalog file;

comparing first make and model attributes from the catalog file to second make and model attributes of the device; and

wherein a mismatch between the first and the second make and model attributes provides an indication that the image is not a properly released image for the device.

19. The method of claim 18, further comprising the step of determining an operational mode of the device, and wherein the step of blocking the use of the image to boot the device is bypassed when the operational mode is set to test mode.

20. The method of claim 11, further comprising the step of loading the image into random access memory (RAM) of the device, and wherein the step of examining is processed after the step of loading.

21. The method of claim 11, wherein when the step of examining determines that the image is a properly released image, the method further comprising the steps of:

erasing a previous image from flash memory of the device;

programming the flash memory of the device with the properly released image.

22. The method of claim 11, wherein the step of examining comprises the steps of:

creating a first hash of the image;

extracting a second hash of the image from the catalog file;

comparing the first hash and the second hash;

extracting a signature certification from the catalog file;

validating the signature certification; and

extracting first make and model attributes from the catalog file;

comparing first make and model attributes from the catalog file to second make and model attributes of the device; and

wherein any one of a first mismatch between the first hash and the second hash, a failure of the step of validating the signature certification, and a second mismatch between the first and the second make and model attributes provides an indication that the image is not a properly released image for the device.

23. A portable computing device, comprising:

flash memory, the flash memory including a protected area and an unprotected area;

a bootloader stored in the protected area of flash memory, the bootloader containing a crypto module;

an operating system image stored in the unprotected area of flash memory;

random access memory (RAM); and

wherein the crypto module of the bootloader is operative to examine an image update to determine if the image update should be programmed into the unprotected area of flash memory to boot the device based on information included in a signed catalog file embedded in the image update.

24. The device of claim 23, wherein the crypto module programs the image update into the unprotected area of flash memory when the device is in test mode.

25. The device of claim 23, wherein the bootloader stores the image update in the RAM until the crypto module determines that the image update should be programmed into the unprotected area of flash memory to boot the device.

26. The device of claim 25, wherein the crypto module calculates a first hash of the image update, extracts a second hash from the catalog file, and compares the first hash and the second hash, the crypto module blocking use of the image update when the first hash and the second hash do not match.

27. The device of claim 25, wherein the crypto module extracts a signature certification from the catalog file and attempts to validate the signature certification, the crypto module blocking use of the image update when the signature certification cannot be validated.

28. The device of claim 25, wherein the crypto module extracts make and model attributes from the catalog file and compares them to make and model information for the device, the crypto module blocking use of the image update when the make and model attributes of the image update do not match the make and model attributes of the device.

29. The device of claim 25, wherein the bootloader erases a current device image from the unprotected area of flash memory and programs the image update into the unprotected area of flash memory when the crypto modules determines that the image update may be used to boot the device.

30. The device of claim 23, wherein a second crypto module of a Mira shell is operative upon a reset of the device to examine the installed operating system image to determine if the installed operating system image should be used to boot the device based on information included in a signed catalog file embedded in the installed operating system image.

31. The device of claim 30, wherein the crypto module in the Mira shell calculates a first hash of the install operating system image, extracts a second hash from the catalog file, and compares the first hash and the second hash, the crypto module in the Mira shell blocking use of the image update when the first hash and the second hash do not match.

32. The device of claim 30, wherein the crypto module in the Mira shell extracts a signature certification from the catalog file and attempts to validate the signature certification, the crypto module in the Mira shell blocking use of the installed operating system image when the signature certification cannot be validated.

33. The device of claim 30, wherein the crypto module in the Mira shell extracts make and model attributes from the catalog file and compares them to make and model information for the device, the crypto module in the Mira shell blocking use of the installed operating system image when the make and model attributes of the installed operating system image do not match the make and model attributes of the device.

34. The device of claim 30, wherein the crypto module in the Mira shell allows the installed operating system image to be used to boot the device when the device is in test mode.